LAB 07

Arpana M Ramaswamy
23/11/2020
Arpana

Q> WAP for distance vector Algorithm

```
class Network:

    def __init__(self, n):
        self.matrix=[]
        self.n=n
    def add_edge(self, s,d,w):
        self.matrix.append([s,d,w])

    def print_solution(self, dist, src):

        print("Vertex table of {}".format(chr(ord('A')+src)))
        print("{0}\t{1}".format("Dest", "Distance"))

        for i in range(self.n):
            print("{0}\t\t{1}".format(chr(ord('A')+i), dist[i]))

    def bellman_ford(self, src):
        dist=[999]*self.n
        dist[src]=0
```

ARPANA M RAMACHANDRAN
23/11/2020
DATE: Arpana PAGE:

```
    for _ in range (self.n-1):
        for s,d,w in self.matrix:
            if dist[s] != 999 and dist[s]+w<dist[d]:
                dist[d] = dist[s]+w

    self.printsolution(dist,src)

matrix=[]
 n = int(input("Enter number of nodes"))
 print("Enter cost matrix:")
 for i in range(n):
     temp = list(map(int, input().split(" ")))
     matrix.append(temp)
 nw = Network(n)
 for i in range(n):
     for j in range(n):
         v = matrix[i][j]
         nw.addedge(i,j,v)

for i in range(n):
    nw.bellman_ford(i)
```